

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#)

Search Forms
Search Results
Help ☐ [Generate Collection](#)
User Searches
Preferences 15 Entries 1 of 6
Logout

File: PGPB

Jan 17, 2002

DOCUMENT-IDENTIFIER: US 20020007358 A1

TITLE: ARCHITECURE OF A FRAMEWORK FOR INFORMATION EXTRACTION FROM NATURAL LANGUAGE DOCUMENTS

Abstract Paragraph:

A framework for information extraction from natural language documents is application independent and provides a high degree of reusability. The framework integrates different Natural Language/Machine Learning techniques, such as parsing and classification. The architecture of the framework is integrated in an easy to use access layer. The framework performs general information extraction, classification/categorization of natural language documents, automated electronic data transmission (e.g., E-mail and facsimile) processing and routing, and plain parsing. Inside the framework, requests for information extraction are passed to the actual extractors. The framework can handle both pre- and post processing of the application data, control of the extractors, enrich the information extracted by the extractors. The framework can also suggest necessary actions the application should take on the data. To achieve the goal of easy integration and extension, the framework provides an integration (outside) application program interface (API) and an extractor (inside) API. The outside API is for the application program that wants to use the framework, allowing the framework to be integrated by calling simple functions. The extractor API is the API for doing the actual processing. The architecture of the

Application Filing Date:

19980901

Summary of Invention Paragraph:

[0007] The natural language documents of a business or institution represents a substantial resource for that business or institution. However, that resource is only a valuable as the ability to access the information it contains. Considerable effort is now being made to develop software for the extraction of information from natural language documents. Such software is generally in the field of knowledge based or expert systems and uses such techniques as parsing and classifying. The general applications, in addition to information extraction, include classification and categorization of natural language documents and automated electronic data transmission processing and routing, including E-mail and facsimile.

Summary of Invention Paragraph:

[0010] According to the invention, there is provided an architecture of a framework for information extraction from natural language documents which is integrated in an easy to use access layer. The framework performs general information extraction, classification/categorization of natural language documents, automated electronic data transmission (e.g., E-mail and facsimile) processing and routing, and parsing.

Detail Description Paragraph:

[0022] The information extraction framework 14 includes preprocessor modules 141 that receive as input the "raw" text from the input access 12 and output "cleaned" text, possibly with additional technical information. This can involve stripping of

irrelevant pieces of text (like technical mail headers), filtering out special characters of tags or converting between different character sets. This is done inside the framework using flexible, configurable preprocessor modules, that can be extended by user built preprocessor libraries in exactly the same fashion user built extractors can be integrated.

CLAIMS:

8. The architecture of claim 7, wherein the preprocessing means includes at least one (i) stripping means for stripping irrelevant pieces of text, (ii) filter means for filtering out special characters of tags and (iii) converting means for converting between different character sets.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)

☐ **Generate Collection**

L5: Entry 2 of 6

File: USPT

Apr 6, 2004

DOCUMENT-IDENTIFIER: US 6718367 B1

TITLE: Filter for modeling system and method for handling and routing of text-based asynchronous communications

Application Filing Date (1):
19990601

Brief Summary Text (14):

The following publications and standards provide additional information into the background of the arts of e-mail routing, natural language processing, and pattern recognition: 1. Internet Network Information Center ("InterNIC") Request for Comment 821, "Simple Mail Transfer Protocol" (SMTP), Filename RFC821.TXT from <http://www.internic.net>. 2. International Telecommunications Union ("ITU") Recommendation X.400, available from the ITU, Berne, Switzerland, and from the ITU's website at www.itu.org. 3. "Fuzzy and Neural Approaches in Engineering" by Lofteri H. Tsoukalas and Robert E. Uhrig, published by John Wiley and Sons, Inc., copyright 1997, ISBN number 0-47116-003-2. 4. "Pattern Recognition and Image Analysis" by Earl Gose, Richard Johnsonbaugh, and Steve Jost, published by Prentice Hall, copyright 1996, ISBN number 0-13-23645-8. 5. "Natural Language and Exploration of an Information Space: The ALFresco Interactive System", a white paper by Olivero Stock, appearing starting on page 421 of the book "Readings in Intelligent User Interfaces", edited by Mark T. Maybury and Wolfgang Wahlster, published by Morgan Kaufman Publishers, Inc., copyright 1998, ISBN number 1-55860-444-8. 6. U.S. Pat. No. 5,768,505 to Gilchrist, et al. 7. U.S. Pat. No. 5,859,636 to Pandit.

Detailed Description Text (4):

The system and method yields tagged e-mails, in which the tags contain the relative scores or rankings of these properties, and further ranks each message within a general property category to sub-properties. The tagged e-mails can then be routed for review by one or more appropriate corporate divisions, departments, or individuals, or a reply could be automatically generated.

Detailed Description Text (12):

The tagged and characterized e-mail messages (7) are then output by the filter and modeler via a number of common data transfer means (6), including all of the means listed for receiving the message input described previously.

CLAIMS:

27. The system for filtering and modeling electronic text messages of claim 1 wherein said clustering means further comprises a k-means means for producing message tags in the message tag set.

28. The system for filtering and modeling electronic text messages of claim 1 wherein said clustering means further comprises a isodata means for producing message tags in the message tag set.

29. The system for filtering and modeling electronic text messages of claim 1

wherein said clustering means further comprises a backpropagation learning analysis means for producing message tags in the message tag set.

30. The system for filtering and modeling electronic text messages of claim 1 wherein said message tag set further comprises an author's attitude tag.

31. The system for filtering and modeling electronic text messages of claim 1 wherein said message tag set further comprises an issue-problem tag.

32. The system for filtering and modeling electronic text messages of claim 1 wherein said message tag set further comprises a request tag.

33. The system for filtering and modeling electronic text messages of claim 1 wherein said message tag set further comprises an author's profile tag.

34. The system for filtering and modeling electronic text messages of claim 1 wherein said message tag set further comprises an author's education level tag.

35. The system for filtering and modeling electronic text messages of claim 1 further comprises a learning means which includes: a tagged message reception means for receiving said tagged messages from said clustering means; a network update means which is capable of modifying parameters, thresholds, and coefficients within said feature extraction means and within said clustering means; and a user interface means for presenting the received electronic text message and said message tag set, receiving operator input modifying said message tag set, and providing network updates to the system via said network update means.

48. A process for filtering and modeling electronic text messages of asynchronous communications systems of claim 36 further comprising the steps: presentation of the electronic text message and the message tags to a user via a user interface; receiving corrections to said message tags via said user interface from said user; and automatically modifying logic within said determination of inherent factor within said text message.

49. A computer-readable medium containing a data structure for storing property tags for electronic text-based messages comprising: an identifier link to a received electronic text-based message, an entry for an author's apparent attitude; an entry for an issue raised by the message, an entry for a request made in the message; an entry for a demographic profile indication for the author; and an entry for an estimated education level of the author.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)

☐ [Generate Collection](#)

L6: Entry 2 of 22

File: USPT

Aug 10, 1999

DOCUMENT-IDENTIFIER: US 5937422 A

TITLE: Automatically generating a topic description for text and searching and sorting text by topic using the same

Application Filing Date (1):

19970415

Brief Summary Text (33):

Possible applications of the present invention include: smart human-computer interface for information retrieval (e.g., voice menu interface, reduce dialog in human-computer interaction, human-computer interface for system control, search engine for internet, automated routing of emergency services, interface for medical on-line diagnosis/data retrieval/consulting, interface for legal/financial information retrieval, etc.), document query (e.g., interface for medical on-line diagnosis/data retrieval/consulting, interface for legal/financial information retrieval, keyword indexing for document retrieval, locate portions of interest: within documents, etc.), automated data sorting (e.g., data routing, e-mail sorting, identification of redundant information in databases, etc.), natural language processing (e.g., disambiguate homonyms, stemming, part-of-speech tagging, etc.), post processing to improve machine transcription (e.g., machine recognition of speech, auto dictation, text conversion from an optical character reader, etc.), and multi-lingual processing (e.g., multi-lingual interface, automatic translation, etc.).

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)



Generate Collection

L6: Entry 1 of 22

File: USPT

Nov 18, 2003

DOCUMENT-IDENTIFIER: US 6650890 B1

**** See image for Certificate of Correction ****

TITLE: Value-added electronic messaging services and transparent implementation thereof using intermediate server

Abstract Text (1):

The present invention provides for a centralized, preprocessing electronic messaging solution that performs value-added tasks to electronic messages on behalf of the ISP or the end user, before these messages are delivered to the destination email server. The service can detect and detain damaging or unwanted messages, such as spam, viruses or other junk email messages, and route electronic messages from various sources covering a variety of topics to wired and wireless destinations, apart from the intended recipient email address, in various formats.

Application Filing Date (1):
20000929

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

WEST Search History

[Hide Items](#)[Restore](#)[Clear](#)[Cancel](#)

DATE: Tuesday, September 21, 2004

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>		
<input type="checkbox"/>	L6	I3 NOT I5	22
<input type="checkbox"/>	L5	L3 and (tag or tagged or append or appending)	6
<input type="checkbox"/>	L4	L3 and (mail adj2 box)	0
<input type="checkbox"/>	L3	20001221	28
<input type="checkbox"/>	L2	20001221	28
<input type="checkbox"/>	L1	(e-mail or email or (electronic adj2 mail)) near8 (sort or sorting or route or routing or direct or directing) near8 (language or topic)	50

END OF SEARCH HISTORY

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)

☐ **Generate Collection**

L5: Entry 6 of 6

File: USPT

Apr 3, 2001

DOCUMENT-IDENTIFIER: US 6212532 B1
TITLE: Text categorization toolkit

Application Filing Date (1):
19981022

Brief Summary Text (8):

The natural language documents of a business or institution represents a substantial resource for that business or institution. However, that resource is only a valuable as the ability to access the information it contains. Considerable effort is now being made to develop software for the extraction of information from natural language documents. Such software is generally in the field of knowledge based or expert systems and uses such techniques as parsing and classifying. The general applications, in addition to information extraction, include classification and categorization of natural language documents and automated electronic data transmission processing and routing, including E-mail and facsimile.

Detailed Description Text (25):

The feature definition performs a process which extracts the text to be used for training from an SGML file or other file (can be text from different tags, e.g., TEXT and HEADER). It thereafter extracts the class label(s) and tokenizes the texts. In embodiments, the feature definition also performs stemming, abbreviation expansion, names or term extraction etc., thereby defining the features to be used. The process then computes the feature counts in various ways, such as computing the class and overall counts for the features.

Detailed Description Text (33):

Testing uses the fraction of the input text set aside in the training step. The test program takes a file in the SGML tagged format as input and consults the configuration file (the same used during training). The testing program also does the same feature definition steps (using the same plug-in DLLs) and uses the same counts, filters, weights and merging as defined in the configuration file.18d. The resulting feature table for the document is then processed using the classifier learned in the learning step. The result is a (possibly empty) set of proposed classes for the document, which is then compared with the class(es) annotated for the document in the SGML file.

Detailed Description Text (37):

By way of example, the application step typically is integrated into an application (e.g. a mail routing/answering application like Lotus Notes). In order to accomplish this, a DLL/shared library based classification API is provided. The text from these applications does not have to be in an SGML format, such that an application using any classification API can feed the text directly to the classification engine. The application program feeding the classification API has to, in embodiments, flag sections of the input text consistent with the tags used in training. The actual classification application is using the same source code that is used in the test program.

Detailed Description Text (65):

In preferred embodiments, the toolkit tools uses an SGML inspired text file format for storing and annotating the input files; however, other files are equally contemplated for use with the present invention. It is SGML based in so far as it used SGML like tags (e.g. <TAGNAME> text </TAGNAME>) to mark units of the text. In embodiments, it is not full SGML since a definition file is not required or used.

Detailed Description Text (67):

For training and testing, there must be, in preferred embodiments, a tag with the pre-assigned categories from which the training algorithm should learn the categorization. The default name for that tag is preferably <CLASS>, but the user may equally specify a different default name. If a document is assigned to more than one class, each class label must appear on a separate line. optionally, the toolkit of the present invention may use the content of a tag the user can specify as an identifier for that document.

Detailed Description Text (68):

The user may have the actual text of the document separated within multiple tags. This can be useful since the toolkit of the present invention offers the option to weight, for example, text in the header (or subject lines) more heavily than text in the body of a document.

Detailed Description Text (91):

Where BOOLEAN_TOGGLE is one of: False, True, On, Off, Yes, No, 0, 1. If set to true, documents with more than one value in the filter tag (i.e., with more than one category) are not filtered. (This step is only meant to filter out documents that would make bad negative examples. Categories can be ignored for classification in the feature selection step) Default: true

Detailed Description Text (115):

Besides specifying what entities to count, the user may also specify some information about the input data. Most importantly, the user may have to tell the system the name of the tag that marks the classes and the name(s) of the tag(s) that mark the text content of the document.

Detailed Description Text (131):

2. The document part this token was found in (the name of the tag from the SGML input file).

Detailed Description Text (177):

The Categorization Applier is a command line tool to quickly use rules created in the training phase. To apply the training results to a new document, the same configuration file as in training and testing is consulted. It assumes the input text to be in SGML tagged format. Only the tags that were selected for analysis in the training step will be extracted and used. The applier will apply the rules (or vectors) generated in the training step to the document and print the categories predicted by the rules (vectors) to the screen.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#)

[Generate Collection](#)

L6: Entry 20 of 22

File: EPAB

Aug 2, 2000

DOCUMENT-IDENTIFIER: EP 1024447 A2

TITLE: Routing of E-mails

Abstract Text (1):

A method and system are described for routing incoming e-mails, for example in a customer interaction centre. E-mail addresses are used which are formed from a traditional user name, a novel topic identifier and a traditional domain name, for example of the form ?user_identifier?:?topic- _identifier?@?domain_name?. The topic identifier of a particular e-mail is relevant to the topic of the content of that e-mail. Routing of e-mails is based on the topic identifiers. By using the topic identifiers in the addresses, the e-mails do not necessarily need to be opened in order to route them. A correspondence between the topic identifiers and the addresses of agents best able to deal with those topics is held in a table, which can easily be updated to take account of changes in circumstances. The source addresses in outgoing e-mails are modified to include the topic identifiers, so that reply e-mails to those outgoing e-mails automatically include the topic

identifiers.

Application Date (1):19991123

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)